- OPERATOR –

Operator: Special symbols that perform specific operations on operands, and then return a result.

<table>
<tr><th>Precedence</th><th colspan="2">Type</th><th>Operator</th><th>Description</th><th>Associativity</th></tr>
<tr><td>1</td><td colspan="2">First Operator</td><td>( )</td><td>Function call</td><td>→</td></tr>
<tr><td rowspan="3">2</td><td colspan="2" rowspan="3">Unary Operator</td><td>!</td><td>Logical negation (opposite)</td><td rowspan="3">←</td></tr>
<tr><td>++ x</td><td>Prefix increment: increments the value of x, and then returns the incremented value.

x = 1
y = ++x;
output: x is 2, y is 2</td></tr>
<tr><td>x++</td><td>Postfix increment:
increments the value of x, but returns the original value that x held before being incremented.

x = 1
y = x++;
output: x is 2, y is 1</td></tr>
<tr><td rowspan="3">3</td><td rowspan="15">이항연산자</td><td rowspan="3">Multiplicative Operator
(산술연산)</td><td>--</td><td>Postfix/prefix decrement
(Works the same as the increments)</td></tr>
<tr><td>*</td><td>Multiplication</td><td rowspan="15">→</td></tr>
<tr><td>/</td><td>Division</td></tr>
<tr><td>%</td><td>Modulo: Returns the remainder of the two numbers after division</td></tr>
<tr><td rowspan="2">4</td><td rowspan="2">Additive Operator
(산술연산)</td><td>+</td><td>Addition</td></tr>
<tr><td>-</td><td>Subtraction</td></tr>
<tr><td rowspan="6">5</td><td rowspan="6">Relational Operator
(비교연산)</td><td><</td><td>Less than</td></tr>
<tr><td><=</td><td>Less than (inclusive)</td></tr>
<tr><td>></td><td>Greater than</td></tr>
<tr><td>>=</td><td>Greater than (inclusive)</td></tr>
<tr><td>==</td><td>Equality</td></tr>
<tr><td>!=</td><td>Inequality</td></tr>
<tr><td>6</td><td>Bitwise AND
(비트연산)</td><td>&</td><td>Binary AND Operator copies a bit to the result if it exists in both operands.

a = 0011 1100
b = 0000 1101
(A & B) will give 12 which is 0000 1100</td></tr>
<tr><td>7</td><td>Bitwise OR
(비트연산)</td><td>|</td><td>Binary OR Operator copies a bit if it exists in either operand.

(A | B) will give 61 which is 0011 1101</td></tr>
<tr><td rowspan="2">8</td><td>Logical AND
(논리연산)</td><td>&&</td><td>expression1 && expression2

true only if both expression1 and expression2 are true</td></tr>
<tr><td>Logical XOR</td><td>^</td><td>If only one of the expressions are true
int a =10;
int b = 5;

a >b ^ a = 10 → false
a >b ^ a = b → true</td></tr>
</table>

		Logical NOT	!	The opposite value
				int a =10;
				int b = 5;
				
				(! (a > b)); → false
				(! (a < b)); → true	

9		Logical OR (논리연산)	||	expression1 || expression2 *true if either expression1 or expression2 is true*	
10	Conditional Operator (Ternary Operator)		? :	variable = **Expression ? expression1 : expression2** If the Expression is true, expression1 is assigned to the variable. If the Expression is false, expression2 is assigned to the variable.	
11	Assignment Operator (대입연산)		=	a = b → a = b;	←
			+=	a += b → a = a+ b;	
			-=	a -= b → a = a - b;	
			*=	a *= b → a = a * b;	
			/=	a /= b → a = a / b;	
			%=	a %= b → a = a % b;	